UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/053,205	11/09/2001	Joe Freeman Britt JR.	04676.P012	5511

7590   09/09/2004

Thomas C. Webster
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER
INOA, MIDYS

ART UNIT	PAPER NUMBER
2188	

DATE MAILED: 09/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/053,205	BRITT, JOE FREEMAN
	Examiner	Art Unit	
	Midys Inoa	2188	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>*01 June 2004*</u>.
2a)☐ This action is **FINAL.**   2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) *1-33* is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on <u>*09 November 2001*</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All b)☐ Some * c)☐ None of:
      1.☐ Certified copies of the priority documents have been received.
      2.☐ Certified copies of the priority documents have been received in Application No. _____.
      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
        application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Objections

1.   The objections to claims 10, 28, 29, 31, 32, and 33 have been overcome by the

amendment received on June 1st, 2004. The objection to these claims has been withdrawn.

Claim Rejections - 35 USC § 102

2.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3.   Claims 19-20, 22, 26 and 28-29 are rejected under 35 U.S.C. 102(e) as being anticipated

by Landau (US 6,549,980).

Regarding Claim 19, Landau discloses a method comprising: maintaining a program code

map ("difference file") on a server (Figure 2, disk 44, which is a separate disk and can be placed

remote from the main system) indicating how program code is allocated among a plurality of

non-volatile memory blocks on a data processing device (the difference file contains a collection

of differences and the exacting locations and contents of related blocks that are different in the

target disks from that in the reference disk); and using said program code map ("difference file")

to facilitate modifications to said program code on said data processing device. Since the

difference file identifies errors, it is used to apply the differences to each block in the target disk,

thus facilitating modifications (Column 3, lines 50-57).

Regarding Claim 22, Landau discloses a method wherein maintaining a program code map ("difference file") comprises: maintaining a list of transactions between said server and said data processing device ("master image"); and running an algorithm ("difference comparison") to construct said map using said list of transactions, said algorithm being an algorithm executed on said data processing device (the device in control of the manufacturing process) to store program code (a collection of differences) within said plurality of non-volatile memory blocks (Column 3, lines 9-35).

Regarding Claims 26, Landau discloses a system comprising: a server to transmit program code to a data processing device (**"partitions mirrored by a server operating system"**, Column 3, lines 3-8) and to monitor (1) which program code is stored on said data processing device (taking a master image, Column 3, lines 9-20) and (2) specific areas in a memory space in which said program code is stored on said data processing device ("extracting locations and contents of blocks...", Column 3, lines 9-20), and to transfer additional program code to said data processing device along with storage location data indicating where in said memory said additional program code should be stored (using a difference file to identify errors and apply differences by transmitting difference data to each block in the target disk, thus facilitating modifications, see Column 3, lines 50-57). **Since this system continuously performs a difference comparison to avoid errors in manufacturing process, it is in fact continuously monitoring the system.**

Regarding Claim 20, Landau teaches calculating a collection of differences ("program code patches") through the execution of a difference comparison between the target disk and the

reference disk (Column 3, lines 9-20) and applying the collection of differences to each block of

the target disk ("transmitting said program code patch", Column 3, lines 50-57).

Regarding Claims 28 and 29, Landau teaches storing exacting locations of the blocks

related to each calculated difference ("program patch"). Since the difference file being kept by

the system for the purpose of applying updates to the target disk contains information about more

than one update or difference at a time, it contains storage locations data about one or more

memory blocks (Column 3, lines 9-20).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

5. Claims 1- 4 and 6-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Computer Networks, by Larry L. Peterson and Bruce S. Davie (Peterson et al.) in view of Eidt et

al. (6,219,830)

Regarding Claim 1, Peterson et al. teaches splitting program code into one or more blocks

("data packets"); assigning each of said blocks a header ("individual header") containing a

sequence number ("offset field") identifying which portion of said program code each of said

blocks correspond to (see Figure 4.3). Peterson discloses using packets in the transmission of

data through a network where packets are placed in a special queue when they need low delays

and where packets hop from router to router until a destination is reached. In this transmission

process, it is inherent that the receiving end of each packet; a special queue or a router; stores

the received data (Chapter 4, page 252, paragraph 1; and page 253 paragraph 1). Peterson does

not disclose storing said one or more blocks of program code **and** said associated headers. **Eidt**

et al. discloses a memory array in which both the header section and data sections of an

application program are stored in memory regions of the read-only portions of memory

106 (Column 9, lines 45-58, and Figure 3). Since memory 106 plays the role of main

memory coupled to a CPU and a disk (Figure 1) it must be a non-volatile memory in order

to prevent the system from loosing imperative data during a power loss. It would have been

obvious to one of ordinary skill in the art at the time the invention was made to allow the system

of Peterson et al. to store the header information after the transfer completion in order to allow

the system to identify the data sections to be allocated for access (see Column 9, line 62 –

Column 10, line 2)

Regarding Claim 2, Peterson et al. teaches using the offset field within the header of each

packet to determine the order of the packet within the entire data stream in order to be able to

reassemble the data stream at the receiving end of the network.   The reconstructed code can then

be processed by the receiving system in response to a request from a related application (Chapter

4, pages 253-256 and Figure 4.5).

Regarding Claim 3, Peterson et al. teaches headers including a checksum field ("error

detection"), which is used to determine the validity of each transmitted packet (Chapter 4, page

253).

Regarding Claim 4, Peterson et al. teaches using CRC instead of a checksum as an error

detection method, which computes over the bytes composing a packet and then appends to the

packet by the network hardware. The CRC provides stronger error detection than a checksum (Glossary, page 678 and Chapter 2, pages 92-101).

Regarding Claims 6 and 7, Peterson et al. teaches a cryptography algorithm where a public key is used to encrypt a message that is to be transmitted, and a private key is used to decrypt the message once it has been transmitted. In this algorithm, the verification of the public key through the use of a private key is analogous to the verification of a signature (Chapter 8, pages 570-572 and Chapter 8, page 588).

6.   Claims 5, 8-9, and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Computer Networks, by Larry L. Peterson in view of Bruce S. Davie (Peterson et al.) as applied to claims 1 and 19 above, and further in view of Landau (US 6,549,980).

Regarding Claim 5, Peterson et al. teaches the invention as set forth by Claim 1 above. Peterson et al. does not teach upgrading or modifying the data that has already been transmitted and stored by replacing specified packets. Landau teaches calculating differences between data on the target disk and the reference disk, determining the exacting locations of the related blocks to each difference, and applying each difference ("modification" or "update") to each corresponding block within the target disk ("upgrading program code" or "modifying a portion of said program code", Column 3, lines 9-20 and 50-57). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the updating capabilities of the system of Landau to the system of Peterson et al. since such capabilities would allow the system to apply updates to the data that has already been transferred without having the re-

transmit the entire data stream. Adding these capabilities to the system of Peterson et al. would

allow the system to perform updates at a faster rate since less data needs to be transmitted.

Regarding Claim 8, Peterson et al. teaches the invention as set forth by Claim 1 above.

Peterson et al. also teaches using the offset field within the header of each packet to determine

the order of the packet within the entire data stream in order to be able to reassemble the data

stream at the receiving end of the network (Chapter 4, pages 253-256 and Figure 4.5). Peterson

et al. does not teach upgrading or modifying the data that has already been transmitted and stored

by replacing specified packets. Landau teaches calculating differences between data on the

target disk and the reference disk, determining the exacting locations of the related blocks to

each difference, and applying each difference ("modification" or "update") to each

corresponding block within the target disk ("upgrading program code" or "modifying a portion

of said program code", Column 3, lines 9-20 and 50-57). Furthermore, Landau teaches loading

the blocks of data from the target disk onto a master image ("volatile memory"), determining

which data needs to be modified in the target disk by performing a difference comparison

between partitions of the target disk and the master image, and repairing the image of the disk by

writing the known differences over parts of the master image ("overwriting...said program code

with said program code patch") thus creating a difference file (Column 4, lines 7-17 and Column

3, lines 9-21). It would have been obvious to one of ordinary skill in the art at the time the

invention was made to apply the updating capabilities of the system of Landau to the system of

Peterson et al. since such capabilities would allow the system to apply updates to the data that

has already been transferred without having the re-transmit the entire data stream. Adding these

capabilities to the system of Peterson et al. would allow the system to perform updates at a faster

rate since less data needs to be transmitted. It is understood that upon combining the inventions

of Peterson et al. with that of Landau, the combined invention would also use the offset fields to

identify the location of portions of program code.

Regarding Claim 9, Peterson et al. teaches headers including a checksum field ("error

detection"), which is used to determine the validity of each transmitted packet (Chapter 4, page

253). It is understood that upon the transmission of update packets through Peterson's system,

the checksum element would remain active and thus the validity of the update would be

confirmed.

Regarding Claim 23, Peterson et al. does not teach the use of Flash memory as the non-

volatile memory in the receiving end of the transmitting network. It would have been obvious to

one of ordinary skill in the art at the time the invention was made to replace the ordinary disk in

a receiving system for a Flash memory since it is very portable and thus, would be versatile if the

receiving system is a portable computer or another portable device.

Regarding Claim 24, Landau teaches loading the blocks of data from the target disk onto

a master image ("volatile memory"), determining which data needs to be modified in the target

disk by performing a difference comparison between partitions of the target disk and the master

image, and repairing the image of the disk by writing the known differences over parts of the

master image ("overwriting...said program code with said program code patch") thus creating a

difference file (Column 4, lines 7-17 and Column 3, lines 9-21). Landau does not teach

identifying each portion of the program code with an offset. Peterson et al. teaches using the

offset field within the header of each packet to determine the order of the packet within the entire

data stream in order to be able to reassemble the data stream at the receiving end of the network

(Chapter 4, pages 253-256 and Figure 4.5). It is understood that upon combining the inventions

of Peterson et al. with that of Landau, the combined invention would also use the offset fields to

identify the location of portions of program code.

Regarding Claim 25, Peterson et al. teaches using CRC instead of a checksum as an error

detection method, which computes over the bytes composing a packet and then appends to the

packet by the network hardware. The CRC provides stronger error detection than a checksum

(Glossary, page 678 and Chapter 2, pages 92-101).

7.   Claims 10-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Computer

Networks, by Larry L. Peterson in view of Bruce S. Davie (Peterson et al.) and further in view of

Landau (US 6,549,980).

Regarding Claims 10-14, Peterson et al. teaches the transmission of data through a

network in which the data is divided into packets ("blocks") and each packet contains headers

with offset information indication the location of each packet within the data stream ("indication

blocks in the processing device into which each application is to be stored"). Peterson et al. does

not teach maintaining a list of data transactions on the processing device and using such list to

construct a map of all applications stored on the processing device. Landau teaches a computer

system in which a collection of differences ("modifications") and the exacting locations and

contents of related blocks ("program code map") that are different in the target disks from that in

the reference disk are kept in a difference file (Column 3, lines 9-20). The difference file

("program code map") is used to apply the differences to each block in the target disk

("facilitates modifications", Column 3, lines 50-57). It would have been obvious to one of

ordinary skill in the art at the time the invention was made to apply the updating capabilities of

the system of Landau to the system of Peterson et al. since such capabilities would allow the

system to apply updates to the data that has already been transferred without having the re-

transmit the entire data stream. Adding these capabilities to the system of Peterson et al. would

allow the system to perform updates at a faster rate since less data needs to be transmitted. In

order to benefit from the updating capabilities of the system of Landau et al., the system would

have to create the difference file prior to having any further interaction with the target disk,

regardless of the interactions being new transactions or for the application of updates; creating

the difference file prior would prevent the reapplication of updates that were applied prior to the

difference file ("generating map... prior to new transactions with data processing device").

Regarding Claim 15, Peterson et al. teaches using the offset field within the header of

each packet to determine the order of the packet within the entire data stream in order to be able

to reassemble the data stream at the receiving end of the network (Chapter 4, pages 253-256 and

Figure 4.5).

Regarding Claim 16, Landau teaches loading the blocks of data from the target disk onto

a master image ("volatile memory"), determining which data needs to be modified in the target

disk by performing a difference comparison between partitions of the target disk and the master

image, and repairing the image of the disk by writing the known differences over parts of the

master image ("overwriting... said program code with said program code patch") thus creating a

difference file (Column 4, lines 7-17 and Column 3, lines 9-21). Landau does not teach

identifying each portion of the program code with an offset. Peterson et al. teaches using the

offset field within the header of each packet to determine the order of the packet within the entire

data stream in order to be able to reassemble the data stream at the receiving end of the network

(Chapter 4, pages 253-256 and Figure 4.5). It is understood that upon combining the inventions

of Peterson et al. with that of Landau, the combined invention would also use the offset fields to

identify the location of portions of program code.

Regarding Claim 17, Peterson et al. teaches headers including a checksum field ("error

detection"), which is used to determine the validity of each transmitted packet (Chapter 4, page

253). It is understood that upon the transmission of update packets through Peterson's system,

the checksum element would remain active and thus the validity of the update would be

confirmed.

Regarding Claim 18, upon the transmission of update packets though Peterson's system,

the offset field within the header is used for locating the portion that such update belongs to. In

addition, in applying the patches or updates to the stored data ("storing blocks of program code

containing program code patch to non-volatile memory"), the existing data must be replaced with

the calculated difference ("patch"). See Column 3, lines 50-57.

8.   Claims 21, 27 and 30-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Landau (US 6,549,980) in view of Computer Networks, by Larry L. Peterson and Bruce S. Davie

(Peterson et al.).

Regarding Claims 21 and 27, Landau teaches the invention as set forth by Claim 20

above. Landau does not teach designating the location of a difference through the use of an

offset in memory. Peterson et al. teaches dividing data into packets for transmission and each

packet having a header including offset information, which designates the location of the packet

within the data stream. It would have been obvious to one of ordinary skill in the art at time the

invention was made to modify the system of Landau et al. to identify the location of each

difference with an offset since such method allows for the location information to dependent

upon the base address of the data stream. Therefore, if the data stream were to change its base

address, the difference, which is being located through the use of an offset, would still be located

in the correct place.

Regarding Claim 30, Landau teaches the invention as set forth by Claim 26 above.

Landau does not teach using a wireless network for the transferring of program code. Peterson et

al. teaches the use of a network for transferring files. It would have been obvious to one of

ordinary skill in the art to modify the system of Landau to operate in the network environment of

Peterson et al. since such modification would allow for updates to be transmitted from remote

locations. It is understood that the network of Peterson et al. may be a wireless network since the

protocols being disclosed by Peterson also apply in a wireless environment.

Regarding Claims 31-32, Landau teaches the invention as set forth by Claim 28 and 29

above. Landau does not teach using a CRC value to determine the validity of data. Peterson et

al. teaches dividing data for transmission into packets having headers including a checksum field

for error detection, which is used to determine the validity of each transmitted packet (Chapter 4,

page 253). In addition, Peterson et al. teaches using CRC value instead of a checksum as an

error detection method, which computes over the bytes composing a packet and then appends to

the packet by the network hardware. The CRC provides stronger error detection than a

checksum (Glossary, page 678 and Chapter 2, pages 92-101). It would have been obvious to one

of ordinary skill in the art at the time the invention was made to equip the system of Landau with

the error detection capabilities of Peterson et al. since such abilities would allow the system to

prevent the storage of invalid data. Upon the transmission of update packets through Peterson's

system, the error detection element would remain active and thus the validity of the update

would be confirmed. In Peterson's system, all data transmitted is formatted into packets with

headers, and thus, the CRC code would be present even if the transmission is only an update

transmission.

Regarding Claim 33, Landau teaches the invention as set forth by Claim 29 above.

Landau does not teach updating an application signature usable to authenticate an application

upgraded by a patch. Peterson et al. teaches a cryptography algorithm where a public key is used

to encrypt a message that is to be transmitted, and a private key is used to decrypt the message

once it has been transmitted. In this authentication algorithm, the verification of the public key

through the use of a private key is analogous to the verification of a signature (Chapter 8, pages

570-572 and Chapter 8, page 588). It would have been obvious to one of ordinary skill in the art

at the time the invention was made to revise the system of Landau by adding the authenticating

abilities of Peterson et al. since such authentication would allow for this system to accept updates

from difference sources and thus, authenticate the sources prior to applying the changes.

Response to Arguments

9.   Applicant's arguments with respect to claims 1-4 and 6-7 have been considered but are

moot in view of the new ground(s) of rejection.

Applicant argues that Peterson does not teach storing said one or more blocks of program

code **and** said associated headers . **Eidt et al. discloses a memory array in which both the**

header section and data sections of an application program are stored in memory regions of the read-only portions of memory 106 (Column 9, lines 45-58, and Figure 3).

Applicant's arguments with respect to claims 19 and 26 have been fully considered but they are not persuasive.

Applicant argues that the system of Landau does not teach continually monitoring. **Since this system continuously performs a difference comparison to avoid errors in manufacturing process, it is in fact continuously monitoring the system.**

Applicant further argues that the difference file is not equivalent to a program code map since such map maintains allocation information for the application of patches and upgrades. **However, the difference performs the same function. This file keeps extracting information, which is equivalent to location information as to where the difference was extracted from (allocation) and it is used to apply the differences ("patch") to the disk with the error.**

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Midys Inoa whose telephone number is (571) 272-4207. The examiner can normally be reached on M-F 7:00am - 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mano Padmanabhan can be reached on (571) 272-4210. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Midys Inoa
Examiner
Art Unit 2188

MI

MANO PADMANABHAN
SUPERVISORY PATENT EXAMINER